GUÍA PASO A PASO DE LABORATORIO — MÓDULO 09

Ciberseguridad IoT Industrial, Cifrado MQTTS (`TLS 1.3 :8883`), PKI X.509 (`OpenSSL SAN`), Cortafuegos `UFW` , `WireGuard` y Supervivencia 24/7 (`log2ram` + `Watchdog`)

Curso: Internet de las Cosas (IoT) con Arduino y Raspberry Pi — Ingeniería Mecatrobótica

Módulo: 09 — Seguridad Ciberfísica, Endurecimiento de Servidores Edge y Alta Disponibilidad

---

1. Objetivos de Ingeniería del Laboratorio

- Construir y cablear el **circuito de seguridad física de gabinete y supervisión de hardware** (tanto en protoboard estilo Tinkercad como en placa impresa PCB FR-4 soldada), integrando un microinterruptor de apertura de puerta (*Tamper Switch* `NC`), un selector de llave física de mantenimiento (*Keylock*), aislamiento por optoacopladores `PC817` y el circuito de latido (*Heartbeat*) del procesador.
- Desplegar una **Infraestructura de Clave Pública (`PKI X.509`)** privada de planta con `OpenSSL` (`aprovisionar_pki_x509_openssl_san.sh`), generando una Autoridad Certificadora Raíz (`RSA 4096 bits`) y certificados de servidor/cliente con extensión **Subject Alternative Name (`SAN` - RFC 9525)**.
- Reconfigurar el broker **Eclipse Mosquitto 2.x** para operar exclusivamente con cifrado `TLS v1.3` **en el puerto** `8883` hacia la red local y aplicar una **Lista de Control de Acceso (`ACL`)** bajo el Principio de Mínimo Privilegio.
- Endurecer el sistema operativo Linux de la Raspberry Pi mediante el cortafuegos de estado `UFW` , protección contra fuerza bruta `Fail2ban` y acceso remoto seguro punto a punto mediante `WireGuard` / `Tailscale` sin abrir puertos en el router.
- Garantizar la supervivencia continua `24/7` durante años configurando el disco en memoria RAM `log2ram` (para evitar el desgaste de celdas Flash NAND de la microSD) y activando el **Watchdog Timer de Hardware Broadcom** (`/dev/watchdog` a $15\ \text{s}$).

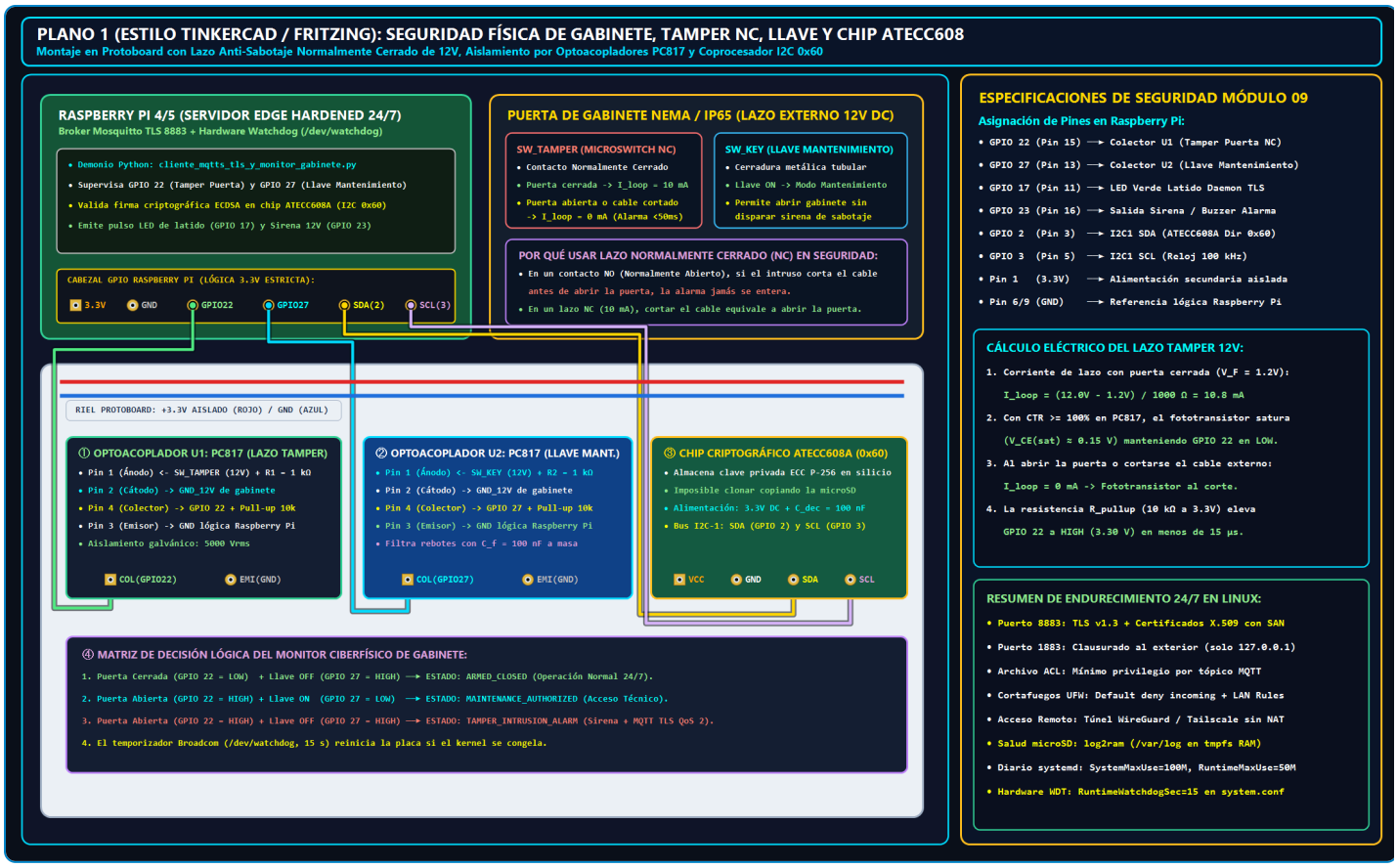
---

2. Fase 1: Montaje del Circuito de Seguridad Física de Gabinete (Tamper + Keylock)

Un servidor Edge en planta cuyos puertos físicos están al alcance de cualquiera sigue siendo vulnerable aunque su red esté cifrada. Por ello implementamos supervisión física de gabinete:

- **Microinterruptor de Sabotaje (*Tamper Switch `SW_TAMPER` *en configuración Normalmente Cerrada* `NC` *)**:
- Mientras la puerta del gabinete metálico permanece cerrada, el contacto `NC` conduce corriente a través de una resistencia limitadora de $1\ \text{k}\Omega$ hacia el LED infrarrojo del primer optoacoplador `PC817` **(** `U1` **)**, manteniendo `GPIO 22` en nivel lógico `LOW` ($0\ \text{V}$).
- Si alguien abre la puerta o corta el cable del sensor, el lazo de corriente se interrumpe instantáneamente y la resistencia *pull-up* a $3.3\ \text{V}$ eleva `GPIO 22` a `HIGH` , disparando una interrupción de intrusión.
- **Interruptor de Llave Mecánica de Mantenimiento (`SW_KEY` en `GPIO 27`)**:
- Permite al ingeniero autorizado poner el gabinete en modo `MAINTENANCE_AUTHORIZED` antes de abrir la puerta, inhibiendo la sirena piezoeléctrica local y registrando en `InfluxDB` una ventana de mantenimiento legítima.

📐 Plano 1: Montaje en Protoboard del Nodo de Seguridad Física de Gabinete



---

3. Fase 2: Aprovisionamiento de Certificados X.509 con Extensión `SAN` (`OpenSSL`)

- Copia el script `aprovisionar_pki_x509_openssl_san.sh` a la Raspberry Pi y ejecútalo indicando la IP estática y el nombre de host local:

  ```bash
  chmod +x aprovisionar_pki_x509_openssl_san.sh
  sudo ./aprovisionar_pki_x509_openssl_san.sh 192.168.1.100 edge-gateway-01.local
  ```

- Verifica que el certificado firmado `server.crt` contenga la extensión `X509v3 Subject Alternative Name`:

  ```bash
  openssl x509 -in /opt/mecatrobotica/certs/server.crt -noout -text | grep -A 2 "Subject Alternative Name"
  ```

- Debe mostrar: `DNS:edge-gateway-01.local, DNS:localhost, IP Address:192.168.1.100, IP Address:127.0.0.1`.

---

4. Fase 3: Cifrado `MQTTS (8883)` y Reglas `ACL` de Mínimo Privilegio

- Genera el archivo de contraseñas cifradas con `PBKDF2-SHA512`:

  ```bash
  sudo mosquitto_passwd -c -b /opt/mecatrobotica/mosquitto/config/passwd bridge_service "ClaveBlindadaBridge2026!"
  sudo mosquitto_passwd -b /opt/mecatrobotica/mosquitto/config/passwd esp32_nodo01 "ClaveNodoCampo2026!"
  ```

sudo mosquitto_passwd -b /opt/mecatrobotica/mosquitto/config/passwd nodered_service "ClaveOrquestador2026!"

```

- Instala el archivo `mosquitto_tls_8883_y_acl_planta.conf` y reinicia el contenedor `iot-mosquitto`.
- **Prueba de Penetración y Auditoría de Reglas `ACL`**:
- Verifica la lectura autorizada con `TLS 1.3`:

  ```bash
  mosquitto_sub -h edge-gateway-01.local -p 8883 --cafile /opt/mecatrobotica/certs/ca.crt \
  -u nodered_service -P "ClaveOrquestador2026!" -t "planta/#" -v
  ```

- Intenta inyectar maliciosamente un cambio de consigna usando las credenciales del sensor exterior `esp32_nodo01`:

  ```bash
  mosquitto_pub -h edge-gateway-01.local -p 8883 --cafile /opt/mecatrobotica/certs/ca.crt \
  -u esp32_nodo01 -P "ClaveNodoCampo2026!" -t "planta/zona1/arduino01/cmd/rpc" -m '{"cmd":"SET_SP","val":45.0}' -q 1
  ```

- **Resultado certificado:** El broker `Mosquitto` rechaza la publicación por violación de `ACL` y el actuador permanece intacto.

---

5. Fase 4: Cortafuegos `UFW`, `log2ram` y `Hardware Watchdog` Broadcom

- Ejecuta el script de endurecimiento de producción:

  ```bash
  chmod +x endurecimiento_ufw_fail2ban_log2ram_watchdog.sh
  sudo ./endurecimiento_ufw_fail2ban_log2ram_watchdog.sh 192.168.1.0/24 100.64.0.0/10
  ```

- Verifica que el temporizador *Watchdog* de hardware del SoC Broadcom esté activo a $15\ s$:

  ```bash
  systemctl show | grep -E "RuntimeWatchdogUSec|RebootWatchdogUSec"
  ```
  ```